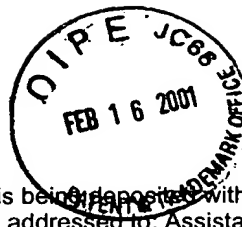
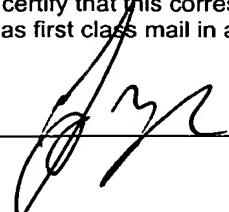


Docket No.: GR 00 P 1031



#4

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231.

By:  Date: February 14, 2001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Xiaoning Nie  
Appl. No. : 09/760,405  
Filed : January 12, 2001  
Title : Data-Processing Device

CLAIM FOR PRIORITY

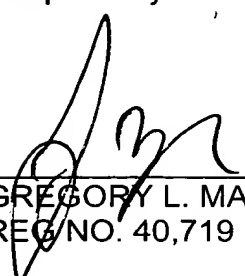
Hon. Commissioner of Patents and Trademarks,  
Washington, D.C. 20231

Sir:

Claim is hereby made for a right of priority under Title 35, U.S. Code, Section 119, based upon the German Patent Application 100 00 960.3 filed January 12, 2000.

A certified copy of the above-mentioned foreign patent application is being submitted herewith.

Respectfully submitted,

  
\_\_\_\_\_  
GREGORY L. MAYBACK  
REG NO. 40,719

Date: February 14, 2001

Lerner and Greenberg, P.A.  
Post Office Box 2480  
Hollywood, FL 33022-2480  
Tel: (954) 925-1100  
Fax: (954) 925-1101

/mjb



## Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

**Aktenzeichen:** 100 00 960.3

**Anmeldetag:** 12. Januar 2000

**Anmelder/Inhaber:** Infineon Technologies AG, München/DE

**Bezeichnung:** Datenverarbeitungsvorrichtung

**IPC:** G 06 F 9/46

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 24. Januar 2001  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

Best Available Copy

## Beschreibung

## Datenverarbeitungsvorrichtung

- 5 Die Erfindung betrifft eine Datenverarbeitungsvorrichtung zum parallelen Verarbeiten von mindestens zwei unabhängigen Prozessen (Threads) gemäß Patentanspruch 1.

10 Der Durchsatz und die Geschwindigkeit einer Datenverarbeitungsvorrichtung bzw. eines Prozessors kann bekanntermaßen durch Pipelining und eine Erhöhung der Taktfrequenz gesteigert werden.

15 Durch eine Erhöhung des Hardwareaufwandes, d.h. eine Vervielfachung von Einheiten des Prozessors, kann ebenfalls die Datenverarbeitungsrate gesteigert werden. Hierzu sind im wesentlichen zwei Konzepte bekannt, nämlich Multiprozessor-Architektur und Parallelismus auf der Befehlsebene des Prozessors (ILP = Instruction Level Parallelism)

20 Heutzutage hat sich ILP als Konzept zum parallelen Verarbeiten weitgehend durchgesetzt. Die Multiprozessor-Architektur ist dagegen aufgrund der komplizierten Interprozessor-Kommunikation weniger erfolgreich.

25 Ein Beispiel für das ILP-Konzept ist die VLIW (Very Large Instruction Word)-Architektur der digitalen Signalprozessoren der C6x-Baureihe der Firma Texas Instruments. Auch die neue IA-64-Architektur der Firmen Intel und Hewlett-Packard basiert auf dem ILP-Konzept. In den beiden vorgenannten Architekturen wird Parallelismus während des Übersetzens des Programmkodes festgestellt. Intel bezeichnet ihre IA-64-Architektur als "Explicit Parallel Instruction Computing" (EPIC). Beim Übersetzen werden einige vorbestimmte Flags zum  
30 Erkennen der Parallelität gesetzt. Dadurch kann der Prozessor einfach feststellen, wann parallel auszuführende Instruktionen im Programmfluß auftreten und entsprechend reagieren.  
35

Ferner ist dies billiger als beispielsweise eine doppelt vorhandenen Hardware wie bei einem vollständigen Parallelismus, der in den superskalaren Architekturen der PowerPC-Prozessoren eingesetzt wird. Dort wird erst während der Programmlaufzeit ein Parallelismus festgestellt, auf den dann entsprechend reagiert wird. Der Gewinn aufgrund von ILP ist allerdings durch die inhärenten Abhängigkeiten der Daten- und Steueroperationen begrenzt. Zur Vermeidung solcher Abhängigkeiten sind aufwendige Vorarbeiten - beispielsweise eine Berücksichtigung von Daten- und Steueroperationsabhängigkeiten bereits beim Programmieren - erforderlich, die wiederum den gesamten Entwicklungsprozeß verteuern.

In dem Aufsatz "Simultaneous Multithreading: Maximizing On-Chip Parallelism" von D.M. Tullsen, S.J. Eggers, H.M. Levy, veröffentlicht auf den Proceedings of 22<sup>nd</sup> Annular Int. Symposium for Computer Architectures, 1995 wird zur Maximierung des Parallelismus auf Chipebene das Konzept des Simultaneous Multithreading vorgeschlagen. Hierzu werden mehrere unabhängige Prozesse, Befehlsfolgen oder Programmflüsse (Threads) an die mehrfach vorhandenen Einheiten eines superskalaren Prozessors in einem Taktzyklus ausgegeben. Ziel des Simultaneous Multithreading ist es, unter Ausnutzung von Parallelismus alle Einheiten eines Prozessors gleichzeitig auszulasten und dadurch unter anderem den Einfluß hoher Latenzzeiten, die durch einen langsamen Speicher verursacht werden, auf die Prozessor-Performance zu verringern.

Ein Single-Chip-Multiprocessor ist in dem Aufsatz "A single-chip Multiprocessor" von L.Hammond, B.A. Nayfeh, K. Olukotun beschrieben, der 1997 in IEEE-Computer auf den Seiten 79 bis 85 veröffentlicht wurde. In diesem Aufsatz wird insbesondere auf die Unterschiede zwischen Simultaneous-Multithreading (SM) und Chip-Multiprocessors (CMP) hinsichtlich der Performance eingegangen. Ferner wird vorgeschlagen, aufgrund der schnell wachsenden Integrationsdichte von integrierten Schal-

tungen eine Multiprozessor-Architektur zum Verarbeiten paralleler Prozesse zu verwenden.

Aufgabe der Erfindung ist es, eine Datenverarbeitungsvorrichtung bzw. einen Prozessor vorzuschlagen, die bzw. der mindestens zwei Prozesse (Threads) parallel ausführen kann, wobei der Hardware-Aufwand verhältnismäßig gering ist.

Diese Aufgabe wird durch eine Datenverarbeitungsvorrichtung mit den Merkmalen von Patentanspruch 1 gelöst. Bevorzugte Ausführungsformen sind den abhängigen Patentansprüchen entnehmbar.

Eine solche Datenverarbeitungsvorrichtung kann beispielsweise in Telekommunikationsanwendungen als Netzwerkprozessor zur Behandlung der Schicht 1 bis 3 des Protokollstapels von LAN-Anwendungen, in ATM (Asynchronous Transfer Mode)-Schaltern, in IP (Internet Protocol)-Routern, in Frame-Relays, die insbesondere auf DSL (Digital Subscriber Line)-Verfahren, Ethernet und Kabel-Modem basieren, vorteilhaft eingesetzt. Insbesondere bei den vorgenannten Anwendungen treten häufig unabhängige Prozesse zur Verarbeitung verschiedener Aufgaben, beispielsweise verschiedener Protokolle, auf. Solche, als Multiple Threads bezeichnete, parallel ablaufende Prozesse sind im Telekommunikationsbereich als grundlegende Anwendungen entsprechend weit verbreitet. In einer programmierbaren IP/ATM-Schnittstelle (Internet Protocol/Asynchronous Transfer Mode-Input/Output-Processing-Unit) laufen beispielsweise unabhängige Prozesse zur Steuerung verschiedener Datenverbindungen oder zum Steuern getrennter Datenschiebeoperationen.

Der Einsatz der erfindungsgemäßen Datenverarbeitungsvorrichtung ist aber nicht auf die vorgenannten Anwendungen beschränkt, sondern kann überall, wo parallele Prozesse auftreten, eingesetzt werden. Beispiele hierzu sind Steuer- und Regelaufgaben oder auch ganz allgemein in Computern, die im Büro oder privat genutzt werden. Sinnvoll ist der Einsatz gera-

de dann, wenn allgemein Prozesse parallel mit hoher Performance abgearbeitet werden müssen.

Die Erfindung betrifft eine Datenverarbeitungsvorrichtung zum parallelen Verarbeiten von mindestens zwei unabhängigen Prozessen (Threads) mit

- einem Programmspeicher, in dem mindestens ein kompiliertes Programm mit einer Vielzahl N unabhängiger Prozesse abgelegt ist, wobei das kompilierte Programm Informationen über Parallelität und eine Vielzahl von Bündeln mit mehreren Befehlen eines Prozesses aufweist;
- einer Verzweigungssteuereinheit, die den Programmspeicher adressiert;
- Register zum Speichern von Flags und Daten, die in Abhängigkeit vom ausgeführten Prozeß umgeschaltet werden; und
- einer Programmflußsteuereinheit, das das Holen von Bündeln aus dem Programmspeicher und die Verzweigungssteuereinheit und das Ausgeben von Befehlen in Abhängigkeit von in den Befehlen enthaltenen und zur Kompilierzeit des Programms eingebauten Informationen steuert.

Die grundlegende Idee der Erfindung liegt darin, daß die Abhängigkeiten von Daten und Befehlen in unabhängigen, parallel laufenden Prozessen geringer sind als in einem einzelnen Programmfluß (Single Sequential Program Flow) zur Parallelverarbeitung. Beim Übersetzen bzw. Kompilieren eines Programmes in Maschinencode wird das Programm auf Parallelität untersucht und mit besonderen Flags bzw. Informationen zur Kennzeichnung für Parallelität versehen.

Die Architektur der Datenverarbeitungsvorrichtung ist hierzu kompatibel zu einer Einzelprogramm-Architektur (Single Thread Architecture). Die parallel auszuführenden Befehle werden in einem Taktzyklus aus einem Programmspeicher geholt. Jedem einzelnen parallelen Prozeß wird eine Priorität zugewiesen. Nach der Holphase (Fetch-Phase), in der die Datenverarbeitungsvorrichtung über eine Verzweigungssteuereinheit (Branch-

Control-Unit) Daten und Befehle aus dem Programmspeicher holt, wird von einer Programmflußsteuereinheit (Flow-Control-Unit) entschieden, welcher Prozeß gemäß der vergebenen Priorität und der bei der Übersetzung gesetzten Flags bzw. eingebauten Informationen über Parallelität zuerst ausgeführt wird.

Die Datenverarbeitungsvorrichtung weist hierzu Register zur Speicherung der Statusvariablen der Parallelprogramme wie beispielsweise Programmzähler (Program-Counter), Registerfiles, ALU (Arithmetic Logic Unit)-Flags, etc. auf. Die Register sind in Abhängigkeit vom zu verarbeitenden Prozeß umschaltbar und dienen im wesentlichen zur Speicherung von Daten des Prozesses (sogenannter Context-Switch).

In einer bevorzugten Ausführungsform der Datenverarbeitungsvorrichtung sind N Befehlspuffer parallel dem Programmspeicher nachgeschaltet und speichern aus diesem ausgelesene Befehle.

Vorzugsweise wird ein Befehlsausgabeselektor von der Programmflußsteuereinheit derart gesteuert, daß dieser Befehle aus den Befehlspuffern ausliest und N Befehle parallel ausgibt.

Für die Parallelverarbeitung können zentrale Einheiten wie Befehlsdekoder (Instruction-Decoder) und Programmzähler in der Verzweigungssteuereinheit entsprechend den aktiven parallelen Prozessen vervielfacht, beispielsweise verdoppelt, werden. Hierzu müssen nicht für jeden parallelen Prozeß zusätzliche Einheiten vorgesehen werden, sondern nur so viele, wie Prozesse gleichzeitig parallel ausgeführt werden sollen. Weist also beispielsweise ein Programm fünf verschiedene parallele Prozesse auf, sind aber immer nur zwei dieser fünf Prozesse aktiv, so genügt eine Verdoppelung der zentralen Einheiten. Im Vergleich zu den aufwendigen ILP-Architekturen wird insgesamt weniger Hardware benötigt. Einen besonderen

Aufbau zur Verarbeitung der parallelen Prozesse weisen die Einheiten zum Holen der Befehle (Befehlspuffer und Befehlsausgabeselektor) und zum Verzweigen (Branch-Control-Unit) auf. Die vorgeschlagene Architektur ist sowohl mit als auch  
5 ohne Programm- und Daten-Cache nutzbar.

Vorzugsweise sind N Befehlsdekoder zum Dekodieren der ausgegebenen Befehle vorgesehen.

10 Besonders bevorzugt sind mindestens zwei Befehlsausführungseinheiten zum Ausführen der N dekodierten Befehle vorgesehen.

Vorzugsweise mindestens sind ferner mindestens zwei Busse vorgesehen, welche die N Befehlsausführungseinheiten mit einem Speicher für Daten verbinden.  
15

Die Befehle eines oder mehrerer Bündel(s) sind besonders bevorzugt parallel ausführbar.

20 Vorzugsweise gibt die Verzweigungssteuereinheit einen Adreßzeiger aus, der ein Bündel adressiert.

Die Verzweigungssteuereinheit weist in einer weiteren bevorzugten Ausführungsform

- 25 - einen ersten Multiplexer und einen zweiten Multiplexer,
- einen Addierer und
- N Programmzähler auf, wobei
- von der Programmflußsteuereinheit die Anzahl der Befehle in einem Bündel dem Addierer zugeführt wird, der einen  
30 Adreßzeiger und die Anzahl der Befehle addiert,
- Adressen für Programmsprünge oder Funktionsaufrufe und eine Prozeßnummer von der Programflußsteuerung dem ersten Multiplexer zugeführt werden,
- mittels des ersten Multiplexers entweder das Ausgangssig-  
35 nal des Addierers oder die Adressen für Programmsprünge oder Funktionsaufrufe in den dem aktiven Prozeß zugeordneten Programmzähler geschrieben wird bzw. werden,



- über den zweiten Multiplexer, der über die zugeführte Prozeßnummer gesteuert wird, der Inhalt des dem gerade aktiven Prozeß zugeordneten Programmzählers als neuer Adreßzeiger ausgegeben wird.

5

Der Programmflußsteuereinheit werden bevorzugt über einen Subbus vom Ausgangsbus des Programmspeichers

- mindestens ein Bit zur Indikation der parallelen Ausführung von Befehlen, und/oder
- 10 - mindestens ein Bit zur Indikation der Länge des folgenden Befehlsbündel, und/oder
- die Indikation eines oder mehrerer NOPs in den Befehlsbündeln, und/oder
- die Priorität der Prozesse der Befehle
- 15 zugeführt.

Ein Prozeß wird vorzugsweise durch Zuweisen einer Prozeßnummer, einer Priorität und einer Speicheradresse, ab welcher der Prozeß im Programmspeicher abgelegt ist, aufgerufen.

20

Vorzugsweise dient die Datenverarbeitungsvorrichtung als Netzwerkprozessor zum Verarbeiten von Schicht 1 bis 7 Protokoll-Stapeln in Anwendungen wie LAN, ATM-Schaltern, IP-Routern, Frame-Relays, die auf DSL, Ethernet oder Kabel-Modems basieren. Insbesondere bei diesen Anwendungen ist eine Parallelverarbeitung von wesentlicher Bedeutung, so daß die erfindungsgemäße Datenverarbeitungsvorrichtung hier besonders vorteilhaft eingesetzt werden kann.

25

- 30 Aus der folgenden Beschreibung eines Ausführungsbeispiels der Datenverarbeitungsvorrichtung in Verbindung mit der Zeichnung wird die Funktionsweise der Aufbau verdeutlicht. In der Zeichnung zeigt

35

Figur 1 ein Blockschaltbild der erfindungsgemäßen Datenverarbeitungsvorrichtung,

Figur 2 ein Blockschaltbild, das detailliert die Verschaltung des Programmspeichers, der Programmflußsteuereinheit und des Befehlsausgabeselektors darstellt,

5 Figur 3 ein Blockschaltbild mit der Verzweigungssteuereinheit,

Figur 4 ein Zustandsdiagramm zur Erklärung der Arbeitsweise der Programmflußsteuereinheit.

10

Im folgenden wird die Verzweigungssteuereinheit als Branch-Control-Unit, die Programmflußsteuereinheit als Flow-Control-Unit, die Befehlsausführungseinheit als Execution-Unit und der Befehlsausgabeselektor als Instruction-Issue-Selector bezeichnet.

15

In Figur 1 ist das Blockschaltbild der Datenverarbeitungsvorrichtung zur parallelen Verarbeitung zweier Prozesse bzw. Threads dargestellt. Ein in einem Programmspeicher 12 abgelegtes Programm wird über einen Adreßzeiger PC0 von einer Branch-Control-Unit BCU 11 (Verzweigungssteuereinheit) adressiert. In der BCU 11 sind wenigstens zwei Programmzähler für die verschiedenen Prozesse bzw. Threads vorgesehen. Diese Programmzähler werden den gerade aktiven Prozessen zugewiesen. Abhängig davon, welcher Prozeß gerade läuft, d.h. von der Datenverarbeitungsvorrichtung bearbeitet wird, wird als Adreßzeiger PC0 der Inhalt des entsprechenden Programmzählers herangezogen.

25

30 Dem Programmspeicher 12 sind zwei Befehlspuffer IA 13 und IB 14 für die zwei parallel zu verarbeitenden Prozesse nachgeschaltet. Die Befehlspuffer speichern die aus dem Programmspeicher ausgelesenen Befehle.

35 Eine Flow-Control-Unit 10 (Programmflußsteuereinheit) steuert sowohl die BCU 11 als auch die Befehlspuffer 13 und 14. Den Befehlspuffern 13 und 14 ist ein Instruction-Issue-Selector

15 nachgeschaltet, über den Befehle aus den Befehlspuffern 13 und 14 auf zwei Befehlsdekoder 16 und 17 gemultiplext werden.

5 Den Befehlsdekodern 16 und 17 sind Register 18 zur Speicherung von Zero- Carry-, Overflow-Flags der parallel laufenden Prozesse vorgesehen. Die Register 18 weisen wenigstens zwei Registerfiles zum Speicher von Daten und von Zuständen der aktiven Prozesse auf.

10 Dem Register 18 wiederum sind zwei Execution Units EX1 19 und EX2 20 (Befehlsausführungseinheiten) nachgeschaltet. Diese beiden Einheiten dienen zum Ausführen der Befehle. Hierzu sind beide Einheiten EX1 19 und EX2 20 jeweils mit zwei Bussen BUS1 21 und BUS2 22 verbunden, über die auf einen Speicher 23 zugegriffen wird, in dem Daten abgelegt sind. Der Speicher 23 ist vorzugsweise ein Schreib/Lese-Speicher (RAM).

Im folgenden wird die Funktionsweise der oben beschriebenen Anordnung erläutert:

20

Der Programmcode ist in fester Länge kodiert. Die Programme der Prozesse sind nicht notwendigerweise getrennt, sondern können auch in einem Programm vereint sein. Der Startpunkt eines Prozesses bzw. Threads entspricht einem Sprung, kombiniert mit einer zusätzlichen Funktion zum Setzen einer Prozeß- bzw. Thread-Nummer. Das Format eines solchen Startpunktes lautet wie folgt:

25

RUN Thread\_nr Priority Jump\_Adr

30

Der Prozeß wird also durch den Befehl RUN aufgerufen, wobei noch eine Thread-Nummer Thread\_nr, eine Priorität Priority und die Sprungadresse Jump\_Adr, an der der Prozeßcode im Programmspeicher steht, angegeben bzw. zugewiesen werden.

35

Die Befehle sind immer in Bündeln (Bundles) mit einem oder zwei Befehlen gespeichert. Zum Zeitpunkt der Kompilation wird

der ILP derart untersucht, daß die Bündel mit Befehlen parallel ausführbar sind. Das bedeutet, daß während der Kompilation des oder der Programme(s) untersucht wird, welche Befehle weitgehend unabhängig voneinander sind und demnach parallel ausführbar sind. Zwei unabhängige Befehle werden dann in ein Bündel "verpackt". Ferner können auch die Befehle verschiedener Bündel parallel ausführbar sein. Es sind somit auf jeden Fall die Befehle eines Bündels parallel ausführbar als auch gegebenenfalls die Befehle verschiedener Bündel bzw. verschiedener Prozesse. Jedes Bündel weist ein Flag auf, das die Länge der folgenden Bündels angibt. In Abhängigkeit von diesem Flag wird der Adreßzeiger berechnet. Bei der Adressierung des Programmspeichers 12 über den Adreßzeiger PC0 weist dieser eine Bitbreite auf, mit der die maximale Länge eines Befehlsbündels adressierbar ist.

Das Holen der Befehle aus dem Programmspeicher (Instruction Fetch) erfolgt folgendermaßen: der Programmspeicher 12 wird durch den Adreßzeiger PC0 adressiert (der Adreßzeiger PC0 zeigt hierzu auf die Anfangsadresse eines Bündels); dann wird ein Befehlsbündel in einen der Befehlspuffer 13 bzw. 14 gelesen. Anschließend wird ein weiteres Befehlsbündel ausgelesen und in den anderen der Befehlspuffer 13 bzw. 14 geschrieben. Das zweite Befehlsbündel gehört dabei zu einem anderen Prozeß als das erste Befehlsbündel. Somit können zwei aktive Prozesse bearbeitet werden. Insgesamt können mehr als zwei Prozesse vorhanden sein, aber der Prozessor kann in diesem Ausführungsbeispiel nur zwei aktive Prozesse parallel ausführen. Durch Vervielfachen der entsprechenden Einheiten Befehlspuffer, Befehlsdekoder und Execution-Unit können auch mehr als zwei Prozesse aktiv sein, d.h. parallel verarbeitet werden. Bei jedem Holzyklus werden jedenfalls zwei Befehlsbündel aus dem Programmspeicher 12 in die Befehlspuffer 13 und 14 übertragen.

Über den Instruction-Issue-Selector 15, der Multiplexer-Logik umfaßt, werden aus den Befehlspuffern 13 und 14 Befehle der

Befehlsbündel oder NOPs (No Operations) ausgegeben. Die Befehle oder NOPs werden den Befehlsdekodern 16 und 17 zu Dekodierung zugeführt. Es können entweder zwei Befehle aus einem Befehlsbündel, d.h. zwei Befehle aus dem Befehlspuffer 13 oder 14, oder jeweils ein Befehl aus dem Befehlspuffer 13 und 14 über den Instruction-Issue-Selector 15 ausgewählt werden. Wenn die vom Instruction-Issue-Selector 15 ausgegebenen Befehle NOPs sind, kann der Prozessor in einen Power-Down-Modus gehen.

Die Ausgabe und Auswahl der Befehle aus den Befehlspuffern 13 und 14 sowie den Instruction-Issue-Selector 15 wird von der Flow-Control-Unit 10 gesteuert. Dies wird anhand Figur 2 erklärt. Die aus dem Programmspeicher 12 ausgelesenen Befehlsbündel werden über einen Befehlsbus an die Befehlspuffer 13 und 14 geführt. Über einen Befehlssubbus 25 wird folgende Informationen aus dem Befehlsbündel der Flow-Control-Unit 10 zugeführt:

- Ein Bit zur Indikation der parallelen Ausführung von Befehlen oder ein Bit zur Indikation der Länge des folgenden Befehlsbündel. Im Falle von Programmcode mit fester Länge ist allerdings eine Indikation der Länge nicht nötig.
- Die Indikation eines oder mehrerer NOPs in den Befehlsbündeln, wobei ein NOP durch einen anderen Befehl des anderen Prozesses ersetzt werden kann.
- Den aktuellen Prozeß, d.h. die Nummer der Prozesse über einen Thread-Bus 28.
- Die Priorität der beiden Prozesse.

Die Flow-Control-Unit 10 steuert über einen Fetch\_Ctr-Bus 26 die Befehlspuffer 13 und 14 und über einen Issue\_Select-Bus 27 den Instruction-Issue-Selector 15 an. Die internen Zustände der Flow-Control-Unit 10 weisen die Anzahl der verbleibenden Befehle in den Befehlspuffer 13 und 14 der beiden Prozesse. An Ausgangssignalen gibt die Flow-Control-Unit 10 folgende aus:

- Ein Signal zur Inkrementierung des Programmzählers. Das Signal ist 0 für ein Befehlsbündel mit einem Befehl und 2 für ein Befehlsbündel mit zwei Befehlen.
  - 5 - Über den Fetch\_Ctr-Bus 26 ein Fetch\_Ctr-Signal zum Freigeben der Befehlspuffer 13 und 14.
  - Über den Issue\_Select-Bus 27 ein Issue\_Select-Signal zur Steuerung des Instruction-Issue-Selectors 15.
- 10 Die dem Instruction-Issue-Selector 15 nachgeschalteten Befehlsdekoder 16 und 17 dekodieren die zugeführten Befehle. Die Prozeßnummer und die Priorität des jeweiligen Prozesses werden im Register 18 gespeichert.
- 15 In Figur 3 ist der Aufbau der Branch-Control-Unit 11 dargestellt. Die Branch-Control-Unit 11 weist einen Addierer 30 auf, der den Adreßzeiger PC0 und ein über einen Instruction-Bundle-Bus 35 zugeführtes Signal M addiert. Das Signal M ist entweder 1 für ein Befehlsbündel mit einem Befehl oder 2 mit
- 20 einem Befehlsbündel mit zwei Befehlen. Dadurch wird der aktuelle Programmzählerwert entweder um 1 oder 2 inkrementiert, also abhängig davon, ob ein Befehlsbündel mit einem oder zwei Befehlen gelesen wird. Der Ausgangswert des Addierers 30 wird einem ersten Multiplexer 31 zugeführt. Der erste Multiplexer
- 25 31 schaltet entweder das vom Addierer 30 zugeführte Signal oder über einen Br\_Ctr-Bus 36 zugeführte Signale zum Steuern von Sprüngen und Funktionsaufrufen in einen von zwei Programmzählern 32 und 33. In welchen Programmzähler geschrieben wird, hängt von der über einen Thread-Bus 37 zugeführten Pro-
- 30 zeß-Nummer TNr ab. Jedem der beiden aktiven Prozesse ist ein Programmzähler zugeordnet. Wird also beispielsweise ein Befehlsbündel mit zwei Befehlen des Prozesses Nummer 4 aus dem Programmspeicher gelesen, so ist das Signal M gleich 2 und der erste Multiplexer schreibt einen Wert  $PC0+2$  in den Pro-
- 35 grammzähler PC1 33, der dem Prozeß Nummer 4 zugewiesen ist. Über einen zweiten Multiplexer 34 wird einer der beiden Programmzähler 32 und 33 als Adreßzeiger PC0 ausgegeben. Wieder-

um steuert die Prozeßnummer, welcher der beiden Programmzähler ausgegeben wird.

In Figur 4 ist das Zustandsdiagramm der Flow-Control-Unit 10 dargestellt. Die Flow-Control-Unit weist vier verschiedene Zustände 38 bis 41 auf, die sich jeweils durch verschiedene Werte A und B auszeichnen. Die Werte A und B geben an, wieviele Befehle sich noch in dem Befehlspuffer IA 13 bzw. IB 14 befinden. Bei dem dargestellten Zustandsdiagramm ist die Priorität von Prozeß A größer als die Priorität von Prozeß B.

Bei jedem Zustandsübergang ist in dem Diagramm angegeben, welche Aktionen durchgeführt werden, also Ausgabe eines oder zwei der Werte A und/oder B aus den Befehlspuffern IA 13 und/oder IB 14 durch den Instruction-Issue-Selector 15 oder Laden bzw. Nachladen von einen oder zwei Befehlen aus dem Programmspeicher in die Befehlspuffer IA 13 und/oder IB 14. Es können auch NOPs vom Instruction-Issue-Selector 15 ausgegeben werden. 2A und 2B bedeutet, daß zwei Werte aus dem Befehlspuffer IA bzw. IB ausgegeben werden oder in diesen geladen werden, entsprechend bedeutet 1A und 1B die Ausgabe oder das (Nach)Laden nur eines Wertes.

## Patentansprüche

1. Datenverarbeitungsvorrichtung zum parallelen Verarbeiten von mindestens zwei unabhängigen Prozessen (Threads), mit

- 5 - einem Programmspeicher (12), in dem mindestens ein kompiliertes Programm mit einer Vielzahl N unabhängiger Prozesse abgelegt ist, wobei das kompilierte Programm Informationen über Parallelität und eine Vielzahl von Bündeln mit mehreren Befehlen eines Prozesses aufweist;
- 10 - einer Verzweigungssteuereinheit (11), die den Programmspeicher (12) adressiert;
- Register (18) zum Speichern von Flags und Daten, die in Abhängigkeit vom ausgeführten Prozeß umgeschaltet werden;
- einer Programmflußsteuereinheit (10), die das Holen von
- 15 Bündeln aus dem Programmspeicher (12) und die Verzweigungssteuereinheit (11) und das Ausgeben von Befehlen in Abhängigkeit von in den Befehlen enthaltenen und zur Kompilierzeit des Programms eingebauten Informationen steuert.

20

2. Datenverarbeitungsvorrichtung nach Anspruch 1, dadurch gekennzeichnet, daß N Befehlspuffer (13, 14) parallel dem Programmspeicher (12) nachgeschaltet sind und aus diesem ausgelesene Befehle speichern.

25

- 3. Datenverarbeitungsvorrichtung nach einem der Ansprüche 1 oder 2, dadurch gekennzeichnet, daß ein Befehlsausgabeselektor (15) von der Programmflußsteuereinheit (10) derart gesteuert wird, daß dieser Befehle aus den Befehlspuffern (13, 14) ausliest und N Befehle parallel aus-
- 30 gibt.

- 4. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß N Befehlsdekoder (16, 17) zum Dekodieren der ausgegebenen Befehle vorgesehen sind.
- 35



5. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß mindestens zwei Befehlsausführungseinheiten (19, 20) zum Ausführen der N dekodierten Befehle vorgesehen ist.

5

6. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß mindestens zwei Busse (21, 22) vorgesehen sind, welche die N Befehlsausführungseinheiten (19, 20) mit einem Speicher (23) für Daten verbinden.

10

7. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Befehle eines oder mehrerer Bündel(s) parallel ausführbar sind.

15

8. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Verzweigungssteuereinheit (11) einen Adreßzeiger (PC0) ausgibt, der ein Bündel adressiert.

20

9. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Verzweigungssteuereinheit (11)

25

- einen ersten Multiplexer (31) und einen zweiten Multiplexer (34),
- einen Addierer (30) und
- N Programmzähler (32, 33) aufweist, und
- von der Programmflußsteuereinheit (10) die Anzahl (35) der Befehle in einem Bündel dem Addierer (30) zugeführt wird, der einen Adreßzeiger (PC0) und die Anzahl (35) der Befehle addiert,
- Adressen (36) für Programmsprünge oder Funktionsaufrufe und eine Prozeßnummer (37) von der Programflußsteuerung (10) dem ersten Multiplexer (31) zugeführt werden,
- mittels des ersten Multiplexers (31) entweder das Ausgangssignal des Addierers (30) oder die Adressen (36) für

30

35

Programmsprünge oder Funktionsaufrufe in den dem aktiven Prozeß zugeordneten Programmzähler geschrieben wird bzw. werden,

- über den zweiten Multiplexer (34), der über die zugeführte Prozeßnummer (37) gesteuert wird, der Inhalt des dem gerade aktiven Prozeß zugeordneten Programmzählers (32, 33) als neuer Adreßzeiger (PC0) ausgegeben wird.

10. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß der Programmflußsteuereinheit (10) über einen Subbus (25) vom Ausgangsbus (24) des Programmspeichers (12)

- mindestens ein Bit zur Indikation der parallelen Ausführung von Befehlen, und/oder
  - mindestens ein Bit zur Indikation der Länge des folgenden Befehlsbündel, und/oder
  - die Indikation eines oder mehrerer NOPs in den Befehlsbündeln, und/oder
  - die Priorität der Prozesse der Befehle
- zugeführt werden.

11. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß ein Prozeß durch Zuweisen einer Prozeßnummer, einer Priorität und einer Speicheradresse, ab welcher der Prozeß im Programmspeicher abgelegt ist, aufgerufen wird.

12. Datenverarbeitungsvorrichtung nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Datenverarbeitungsvorrichtung als Netzwerkprozessor zum Verarbeiten von Schicht 1 bis 7 Protokoll-Stapeln in Anwendungen wie LAN, ATM-Schaltern, IP-Routern, Frame-Relays, die auf DSL, Ethernet oder Kabel-Modems basieren, dient.

## Zusammenfassung

## Datenverarbeitungsvorrichtung

5 Die Erfindung betrifft eine Datenverarbeitungsvorrichtung,  
insbesondere einen Netzwerkprozessor zum Verarbeiten von  
Schicht 1 bis 7 Protokoll-Stapeln in Anwendungen wie LAN,  
ATM-Schaltern, IP-Routern, Frame-Relays, die auf DSL, Ether-  
net oder Kabel-Modems basieren, der entsprechend einer Anzahl  
10 von parallel zu verarbeitenden Prozessen Befehlspuffer, Be-  
fehlsdekoder und Befehlsausführungseinheiten aufweist, wobei  
im wesentlichen eine ProgrammflußSteuereinheit die Parallel-  
verarbeitung steuert.

15 (Figur 1)

FIG 1

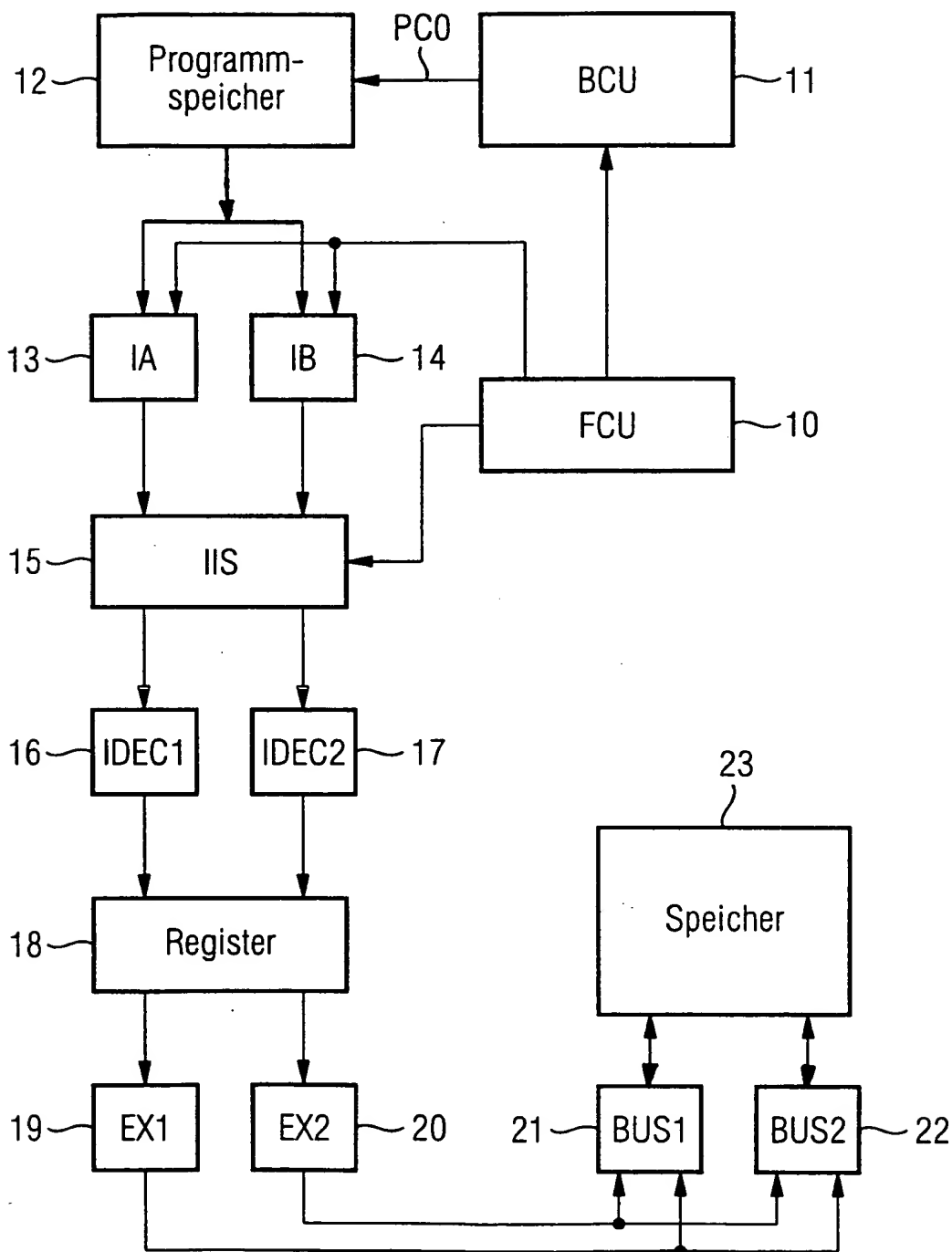


FIG 2

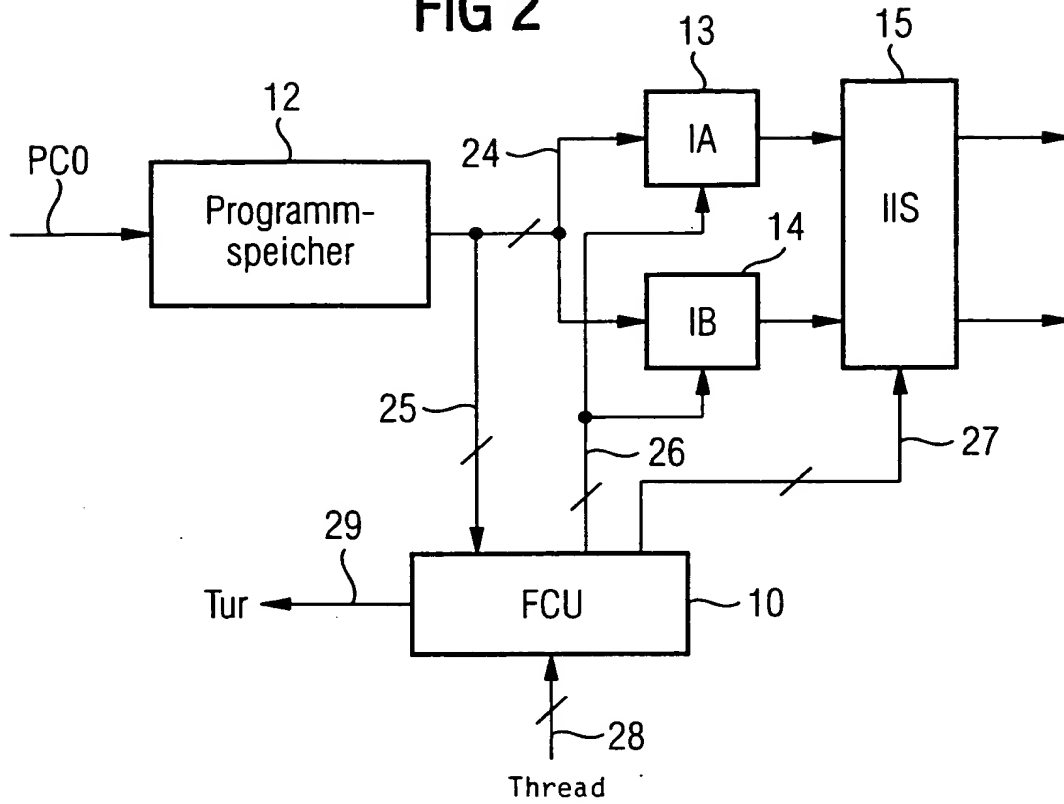


FIG 3

11

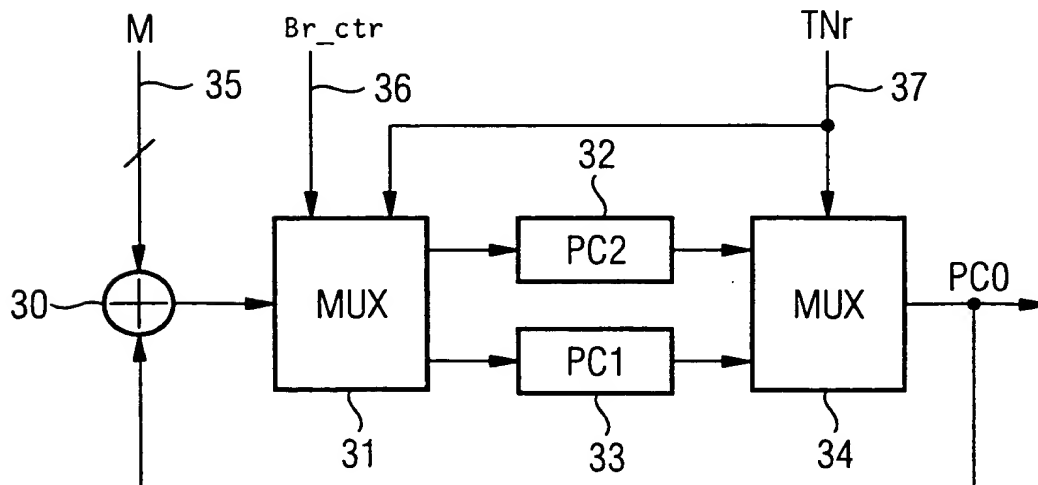


FIG 4

